AMENDMENTS TO THE SPECIFICATION

[0015] FIGs. 10A-G~~C~~ are a sequence of diagrams showing the ROI process of FIG. 9 as applied to an example matrix E.

[0021] The software in memory 103 may include one or more separate programs, each of which comprises an ordered listing of executable instructions for implementing logical functions. In the example of FIG. 1, the software in the memory 103 includes one or more components of the method for recognizing road signs in a digital image 106, and a suitable operating system 107. The operating system 107 essentially controls the execution of other computer programs, such as the method for recognizing road signs in a digital image 106~~108~~, and provides scheduling, input-output control, file and data management, memory management, and communication control and related services.

[0022] The method for recognizing road signs in a digital image 106~~109~~ is a source program, executable program (object code), script, or any other entity comprising a set of instructions to be performed. When a source program, then the program needs to be translated via a compiler, assembler, interpreter, or the like, which may or may not be included within memory 103, so as to operate properly in connection with the operating system 107.

[0025] When the computer 101 is in operation, the processor 102 is configured to execute software stored within the memory 103, to communicate data to and from the memory 103, and to generally control operations of the computer 101 pursuant to the software. The method for recognizing road signs in a digital image 106~~110~~ and the operating system 107, in whole or in part, but typically the latter, are read by the processor 102, perhaps buffered within the processor 102, and then executed.

[0026] When the method for recognizing road signs in a digital image 106~~111~~ is implemented in software, as is shown in FIG. 1, it should be noted that the method for recognizing road signs in a digital image 106~~112~~ can be stored on any computer readable

medium for use by or in connection with any computer related system or method. In the context of this document, a "computer-readable medium" can be any means that can store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, system, or device. The computer-readable medium can be, for example but not limited to, an electronic, magnetic, optical, ~~electromagnetic, infrared,~~ or semiconductor system, system, device, or propagation medium. A nonexhaustive example set of the computer-readable medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM, EEPROM, or Flash memory), and a portable compact disc read-only memory (CDROM). Note that the computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via for instance optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

[0027] In an alternative embodiment, where the method for recognizing road signs in a digital image 106~~113~~ is implemented in hardware, the method for recognizing road signs in a digital image 106~~114~~ can be implemented with any or a combination of the following technologies, which are each well known in the art: a discrete logic circuit(s) having logic gates for implementing logic functions upon data signals, an application specific integrated circuit(s) (ASIC) having appropriate combinatorial logic gates, a programmable gate array(s) (PGA), a programmable gate array(s) (FPGA), *etc.*

[0042] In step 309, a determination is made whether all elements in *R*, *G*, and *B* have been processed. If No, then processing continues at step 310, where the indices *i*, *j* are incremented to advance to the next element in *R*, *G* and *B*. Processing then continues for this next element at step 303. If Yes, then processing stops at step 311~~309~~. At this point, binary segmentation matrices *E1* and *E2* contain 1's at locations that match the color criterion, and 0's at locations

that do not match. These segmentation matrices are used as input to the ROI extraction process (step 203 from FIG. 2).

[0050] V^x (503) then contains one element for each column, with each element having only one of two values: t if the column is valid; or f if the column is invalid. From V^x, the left boundary x_0^1 (504) and right boundary x_0^2 (505) of submatrix S_1 are extracted, so that ~~$S_1 = E(x_0^1, x_0^1; 1, n)$~~ $S_1 = E(x_0^1, x_0^2; 1, n)$.

[0056] Processing continues at step 605~~604~~, where vector C is resampled and then normalized to the same size as the column vector $C_{template}$ of template matrix T. The result is vector C_{norm}.

[0058] Column processing is now complete. At step 608~~604~~, the process waits until the row processing (done by the path starting at step 604~~605~~) is finished. Row processing is similar to column processing. In step 608~~607~~, a vector R is created by summing all elements of S along rows. Thus, R is a vector with one element for each row in S, and the value of that element is the sum of all elements in that row. The formula for computing vector R is:

$$R = \left(\sum_{i=1}^{m} S(i,1), \quad \sum_{i=1}^{m} S(i,2), \quad \dots \quad \sum_{i=1}^{m} S(i,n) \right)$$

[0059] Processing continues at step 609~~608~~, where vector R is resampled and normalized to the same size as the row vector $R_{template}$ of template matrix T. The result is vector R_{norm}. Next, at step 610~~609~~, R_{norm} (rows of the ROI matrix) is correlated with $R_{template}$ (rows of the template matrix). The result is correlation coefficient r_R, which is computed using the following formula:

$$r_R = \frac{\sum_x R_{norm} \times R_{template} - \frac{1}{n}\left(\sum_x R_{norm}\right)\left(\sum_x R_{template}\right)}{\sqrt{\sum_x R_{norm}^{\ 2} - \frac{1}{n}\left(\sum_x R_{norm}\right)^2} \times \sqrt{\sum_x R_{template} - \frac{1}{n}\left(\sum_x R_{ref}\right)^2}}$$

[0060] The two paths (row and column) merge at step 607~~604~~. Step 607~~604~~ continues processing when correlation coefficients for row (r_R) and column (r_R) have been computed. At

step 607~~604~~, the row and column coefficients are compared to coefficient threshold values. In this example embodiment, r_R is compared to a row threshold (0.5), r_C is compared to a column threshold (0.5), and the sum of r_R and r_C is compared to a 2-D threshold (1.2). If all three conditions are met, then a stop sign has been recognized in ROI S, and processing stops at step 612~~610~~. If any condition fails, then no stop sign has been recognized in ROI S, and processing stops at step 611.

[0068] After the correlation coefficient r is computed, processing continues at step 707, where the coefficient r is compared to a coefficient threshold value. If the coefficient is above the threshold, then a stop sign has been recognized in ROI S, and processing stops at step 709~~708~~. If any condition fails, then no stop sign has been recognized in ROI S, and processing stops at step 708~~709~~.

[0095] Processing continues at step 1103. At this step, an OCR (Optical Character Recognition) algorithm is used to recognize the digit represented by each ROI speed limit candidate. OCR algorithms are well known in the art of image processing. In this example embodiment, the algorithm used was developed by Kahan to identify printed characters of any font and size.